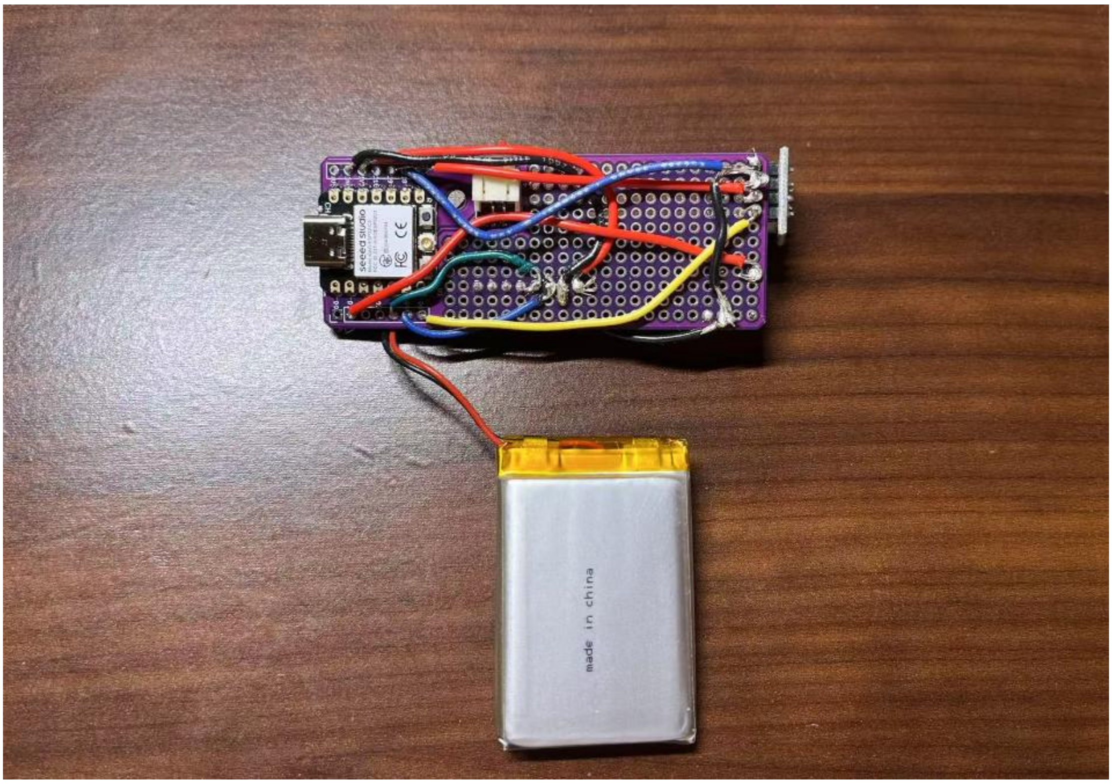
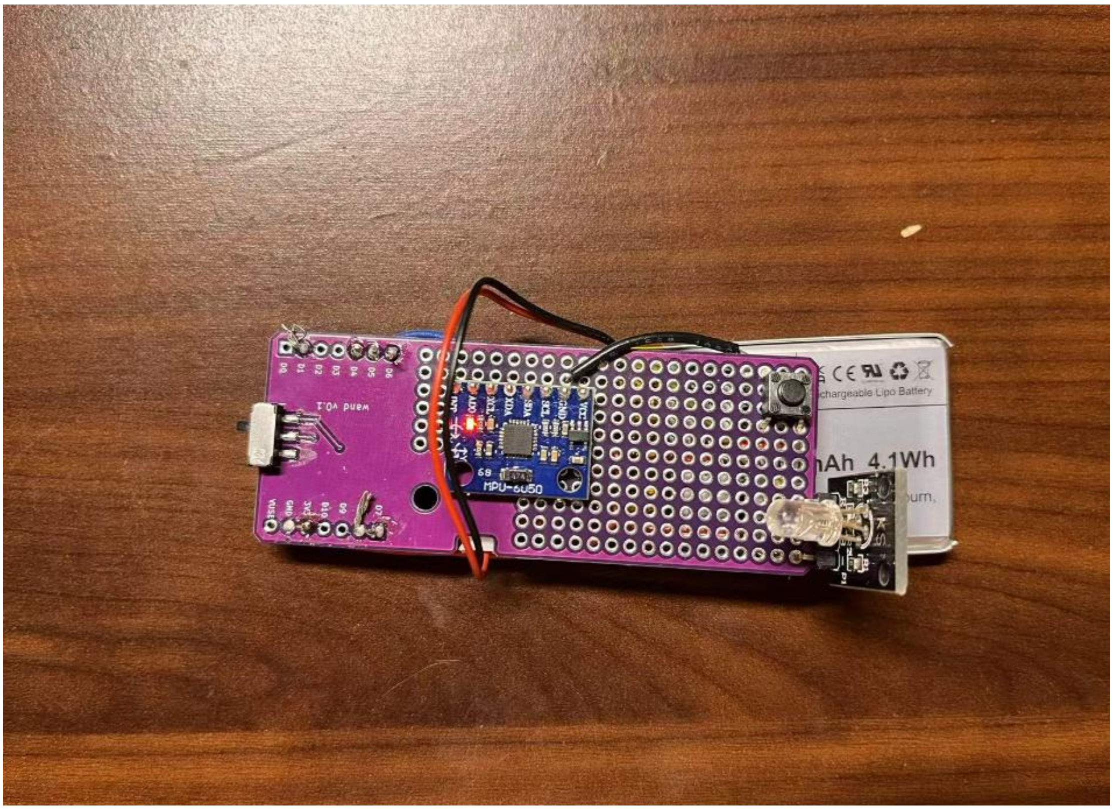
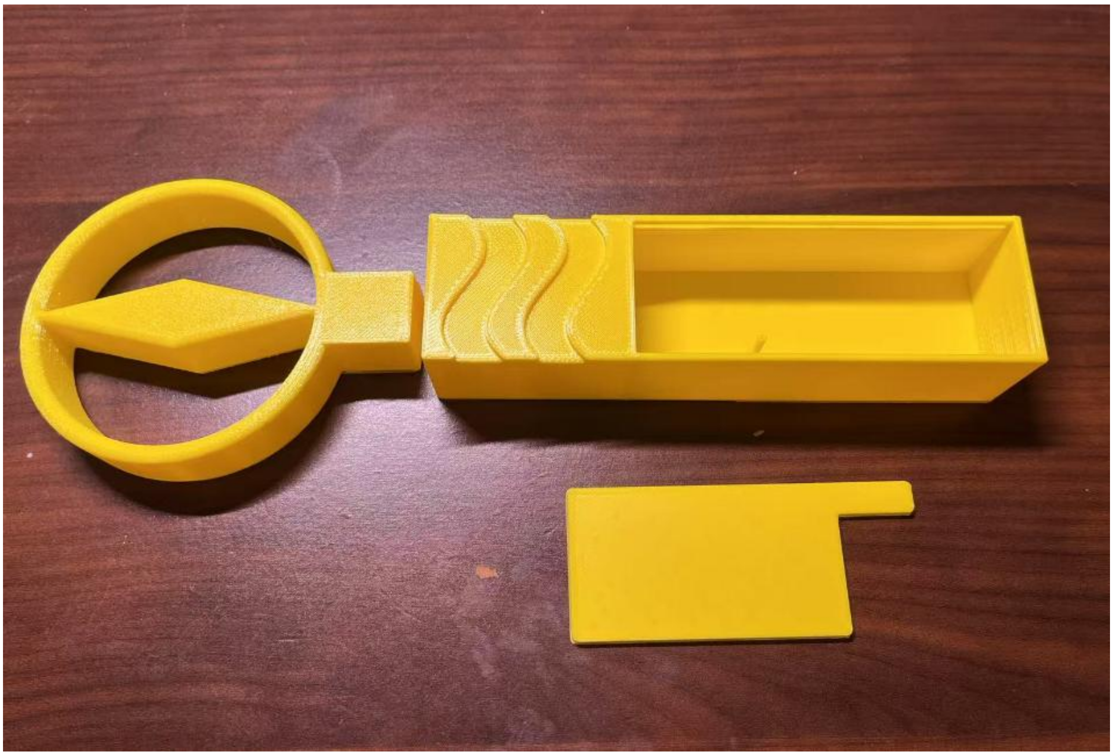
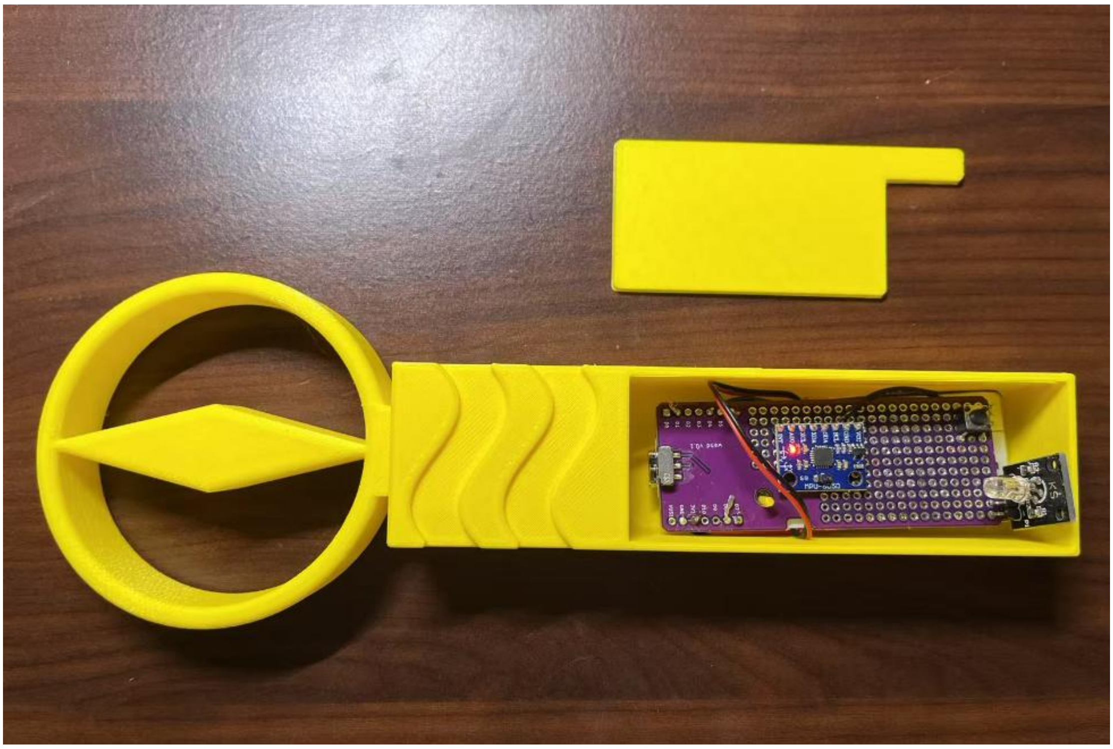
1. Pictures of hardware setup and connections

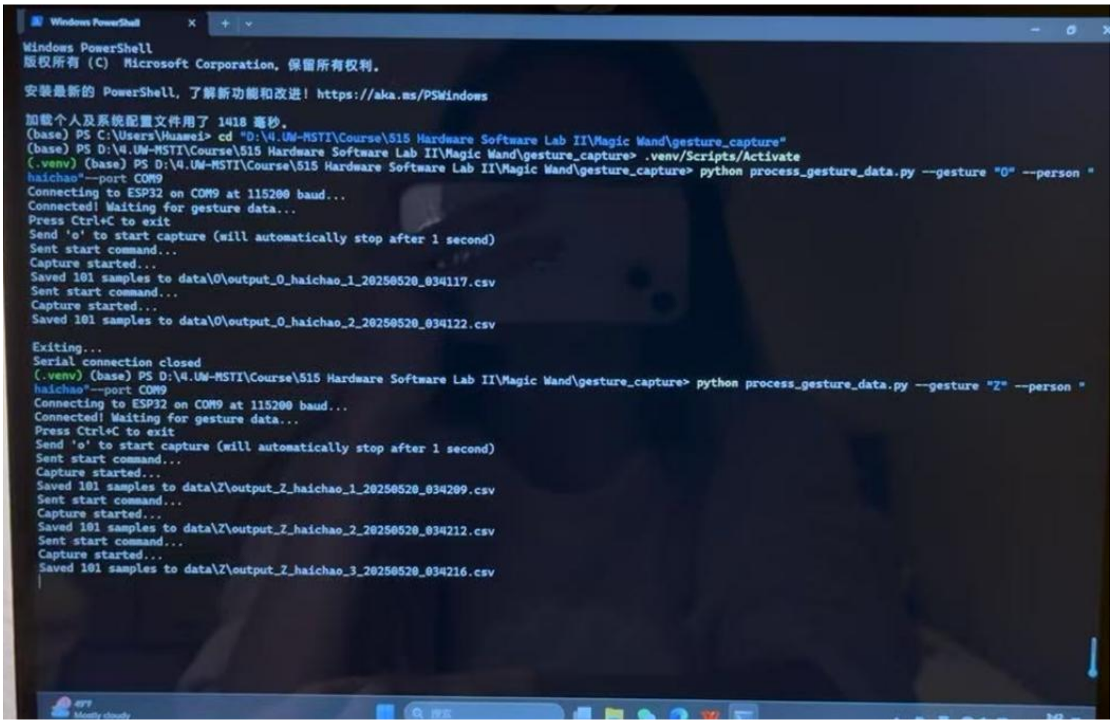


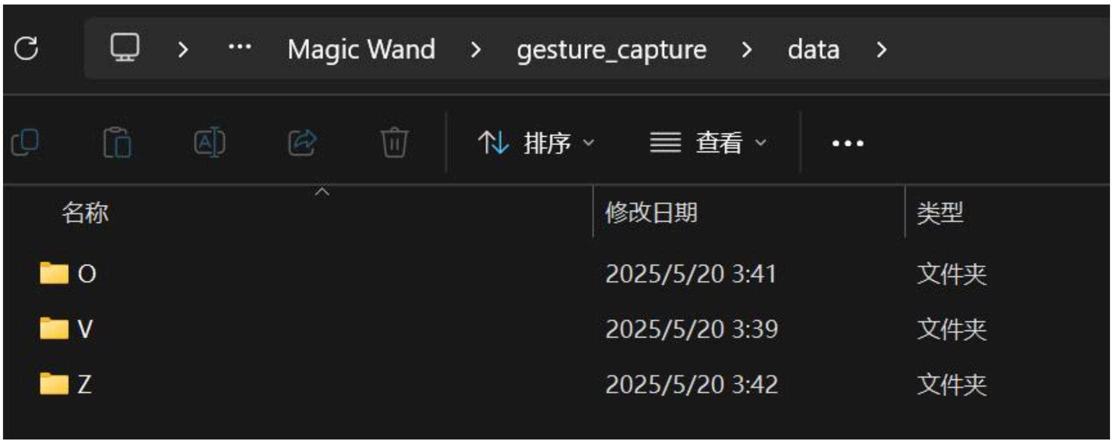


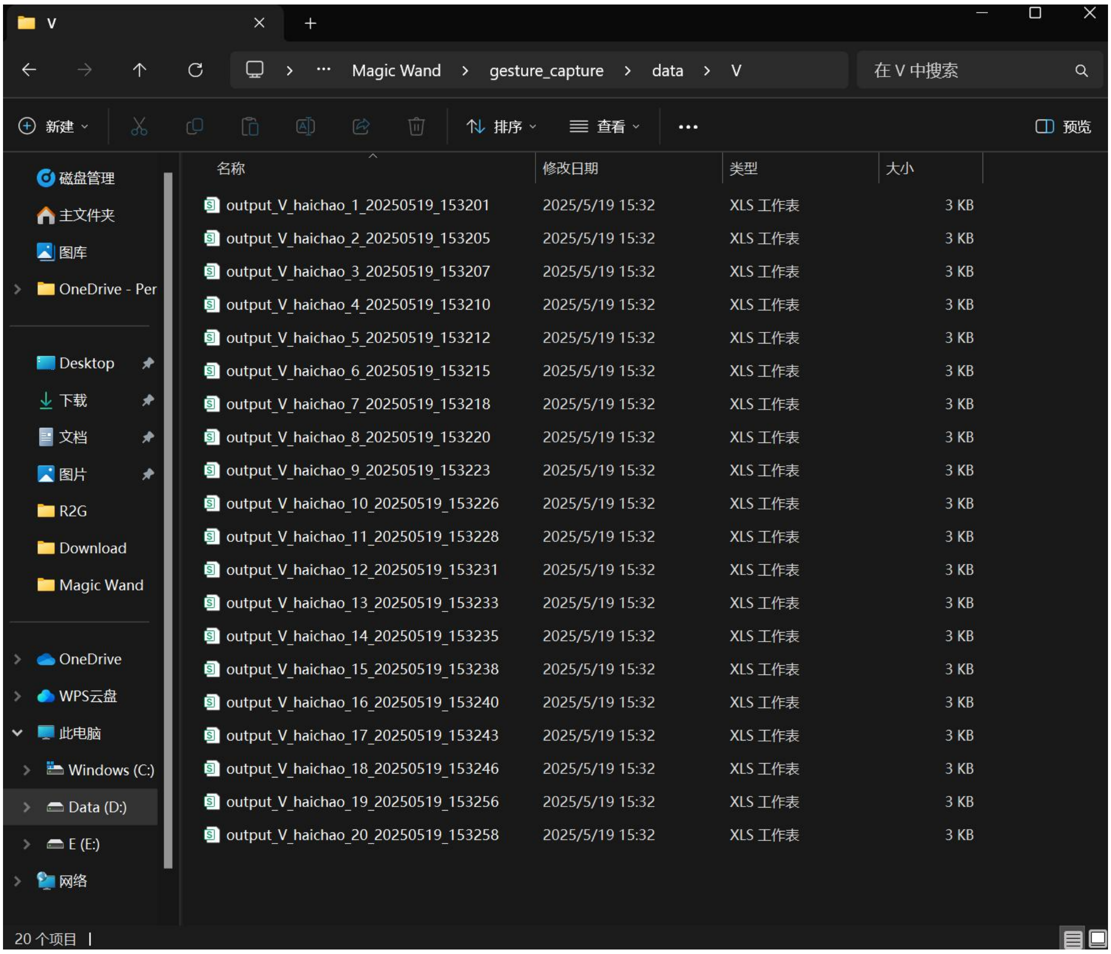


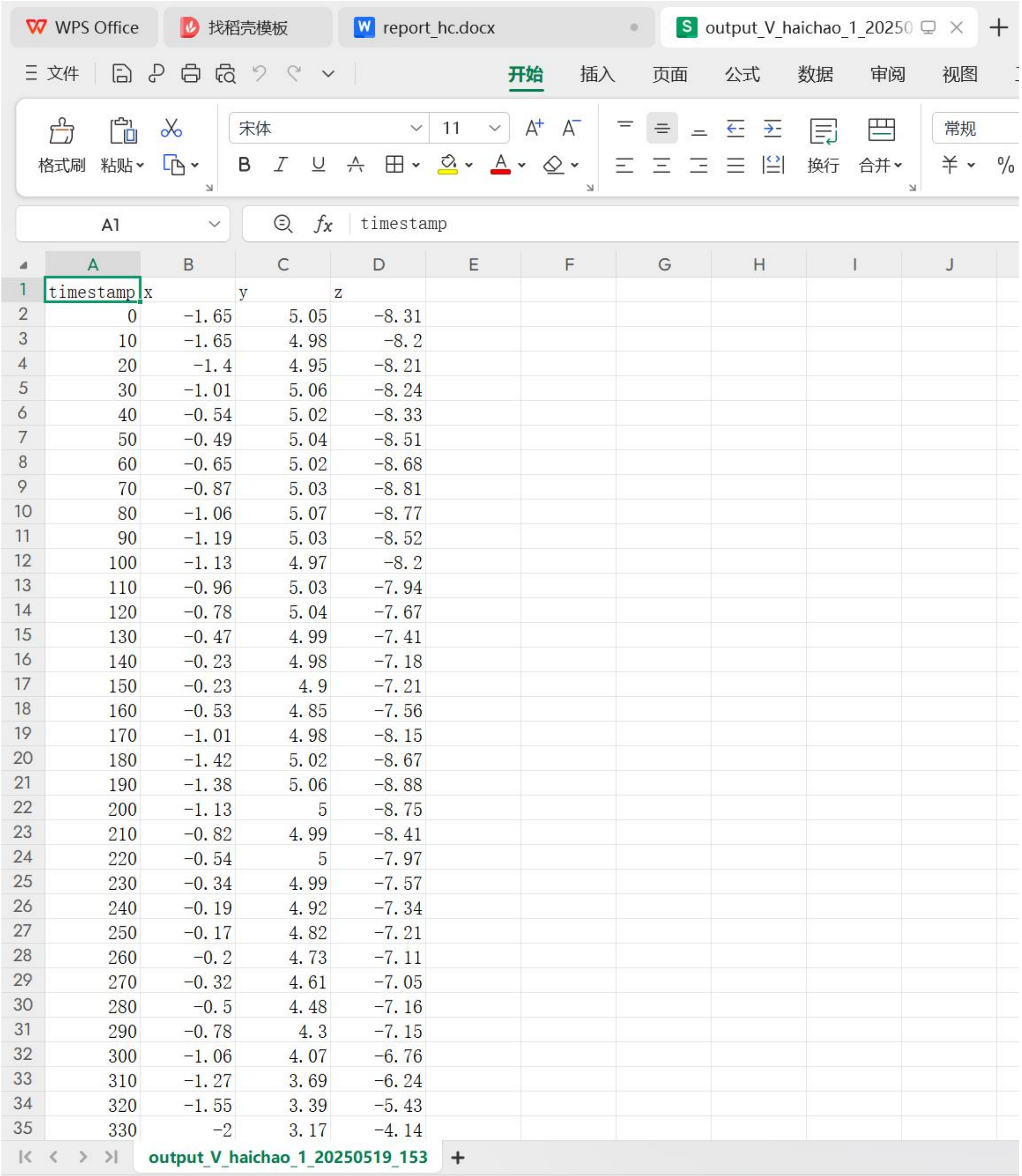


2. Data collection process and results






V
Magic Wand
gesture_capture
data
V
在 V 中搜索
新建
排序
查看
预览
磁盘管理
主文件夹
图库
OneDrive - Per
Desktop
下载
文档
图片
R2G
Download
Magic Wand
OneDrive
WPS云盘
此电脑
Windows (C:)
Data (D:)
E (E:)
网络
名称
修改日期
类型
大小
output_V_haichao_1_20250519_153201 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_2_20250519_153205 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_3_20250519_153207 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_4_20250519_153210 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_5_20250519_153212 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_6_20250519_153215 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_7_20250519_153218 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_8_20250519_153220 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_9_20250519_153223 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_10_20250519_153226 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_11_20250519_153228 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_12_20250519_153231 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_13_20250519_153233 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_14_20250519_153235 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_15_20250519_153238 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_16_20250519_153240 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_17_20250519_153243 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_18_20250519_153246 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_19_20250519_153256 2025/5/19 15:32 XLS 工作表 3 KB
output_V_haichao_20_20250519_153258 2025/5/19 15:32 XLS 工作表 3 KB
20 个项目

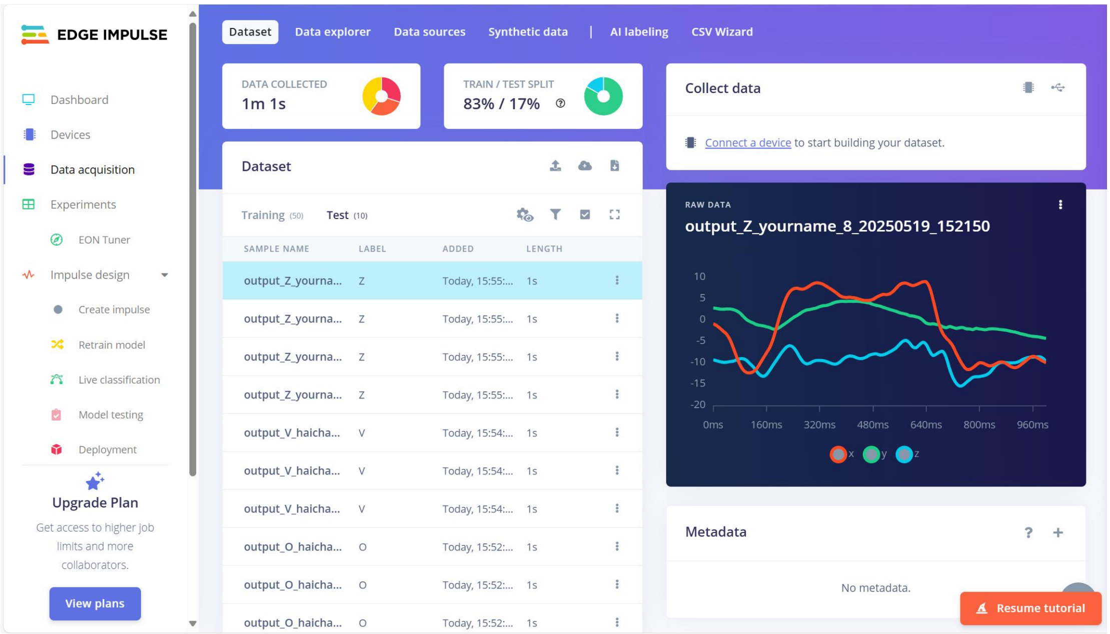


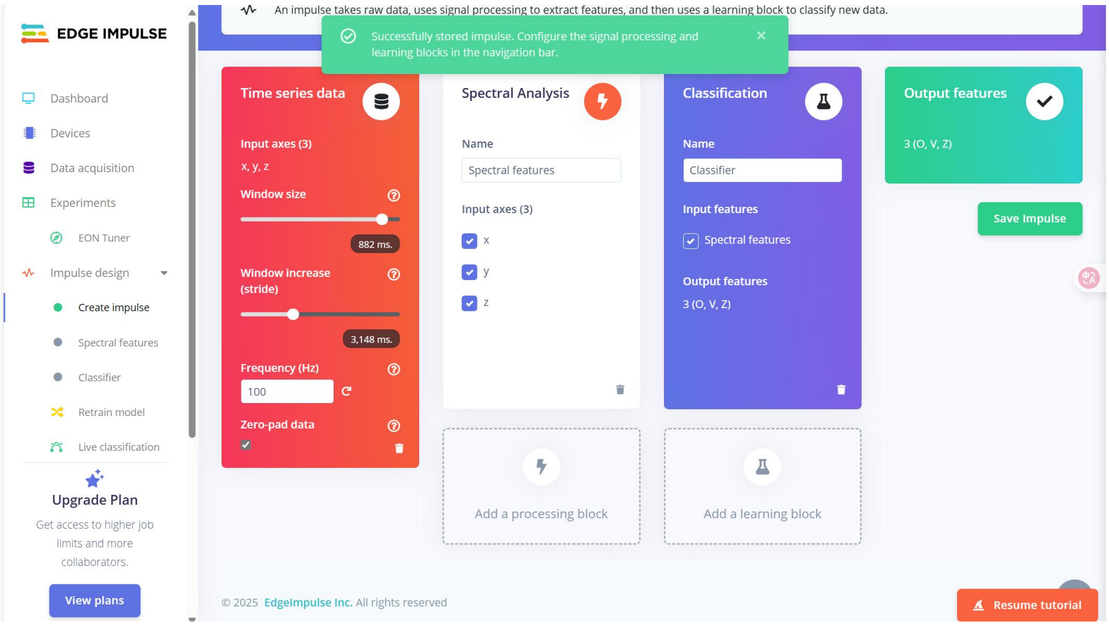
timestamp	x	y	z
0	-1.65	5.05	-8.31
10	-1.65	4.98	-8.2
20	-1.4	4.95	-8.21
30	-1.01	5.06	-8.24
40	-0.54	5.02	-8.33
50	-0.49	5.04	-8.51
60	-0.65	5.02	-8.68
70	-0.87	5.03	-8.81
80	-1.06	5.07	-8.77
90	-1.19	5.03	-8.52
100	-1.13	4.97	-8.2
110	-0.96	5.03	-7.94
120	-0.78	5.04	-7.67
130	-0.47	4.99	-7.41
140	-0.23	4.98	-7.18
150	-0.23	4.9	-7.21
160	-0.53	4.85	-7.56
170	-1.01	4.98	-8.15
180	-1.42	5.02	-8.67
190	-1.38	5.06	-8.88
200	-1.13	5	-8.75
210	-0.82	4.99	-8.41
220	-0.54	5	-7.97
230	-0.34	4.99	-7.57
240	-0.19	4.92	-7.34
250	-0.17	4.82	-7.21
260	-0.2	4.73	-7.11
270	-0.32	4.61	-7.05
280	-0.5	4.48	-7.16
290	-0.78	4.3	-7.15
300	-1.06	4.07	-6.76
310	-1.27	3.69	-6.24
320	-1.55	3.39	-5.43
330	-2	3.17	-4.14

Discussion: Why should you use training data collected by multiple students rather than using your own collected data only? Think about
the effectiveness and reliability of your wand.

Using training data collected by multiple students increases both effectiveness and reliability. It introduces natural variation in how gestures are performed, helping the model generalize better to new users. Relying only on my own data may cause the model to overfit my personal gesture style, reducing performance when others use the wand.

3. Edge Impulse model architecture and optimization





Processing block justification:

I selected Spectra/ Analysis as my processing block because it is well-suited for analyzing repetitive motion data from accelerometers. lt extracts the frequency and power characteristics of the signals over time, which is useful for classifying dynamic gestures like "Z", "O", and "V".

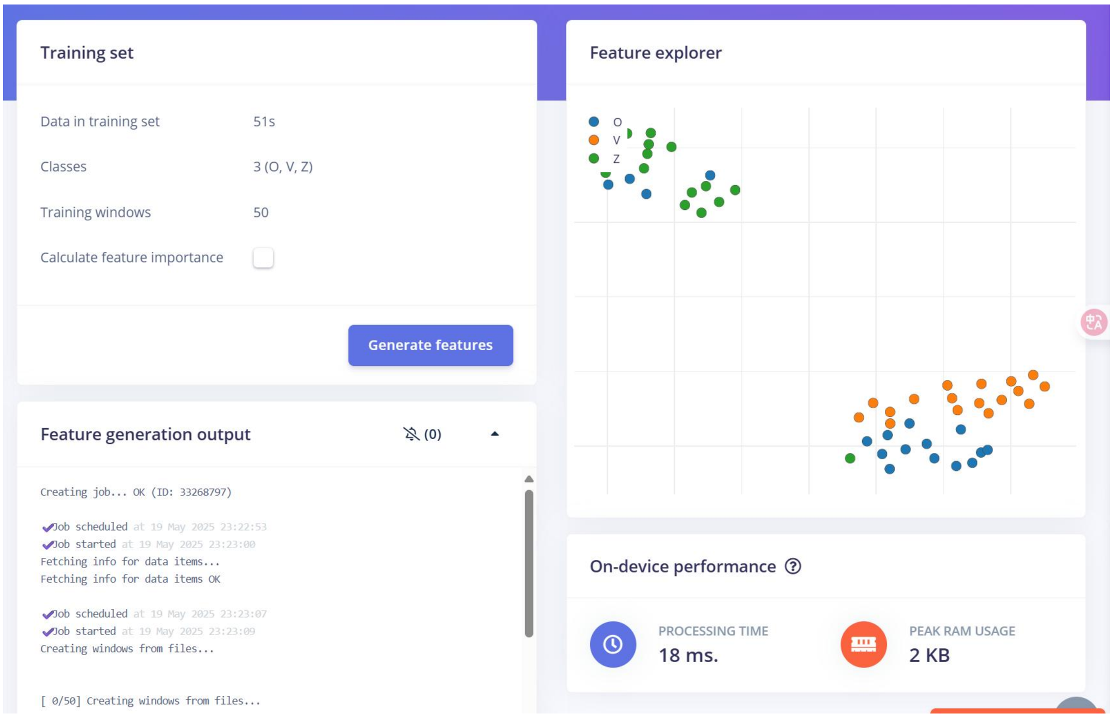
Learning block justification:

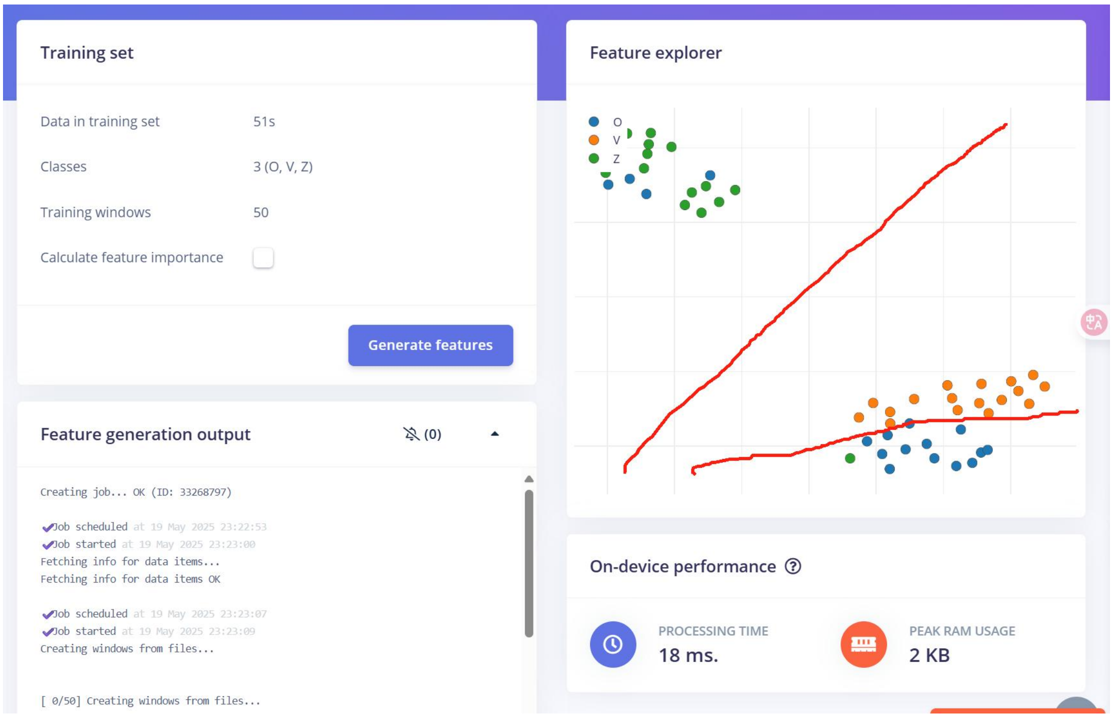
I chose Classification as the learning block since our goal is to distinguish between different gesture types. This is a typical classification task, and the built-in classifier provides a simple and effective way to learn from the spectral features.

Discussion: Discuss the effect of window size. Consider

- the number of samples generated
- the number of neurons in your input layer of neural network
- effectiveness when capturing slow-changing patterns

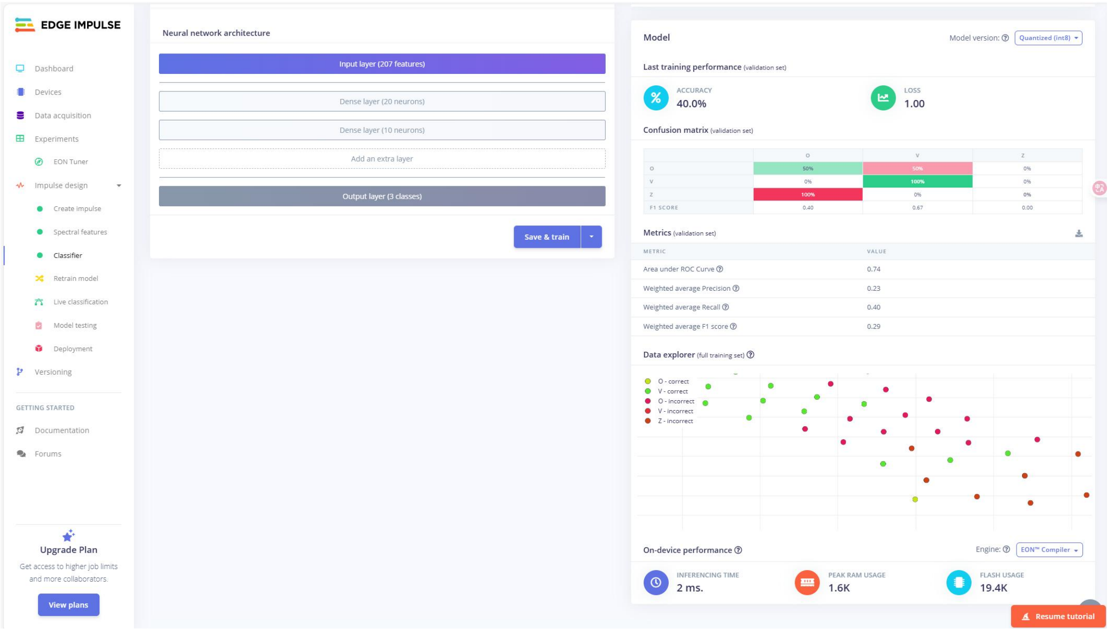
Window size affects how the model sees the gesture. A larger window captures longer patterns, which helps detect slow-changing gestures but reduces the number of training samples and increases the number of input neurons. A smaller window creates more training samples and fewer input neurons, but may miss the full shape of slow gestures.





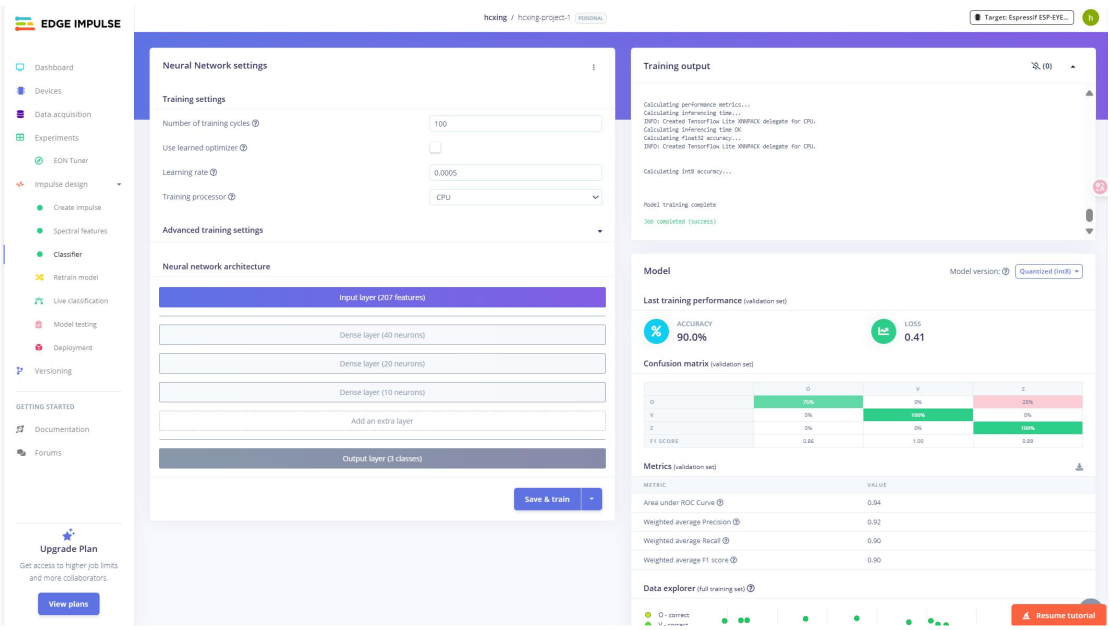
DSP block -- why the generated features are good enough:

The generated features are good enough because the three gesture classes (O, V, Z) are clearly separated in the feature space. Most samples cluster tightly with others of the same label, and there is minimal overlap between clusters. This suggests that the model will be able to learn a clear decision boundary and accurately classify new data.

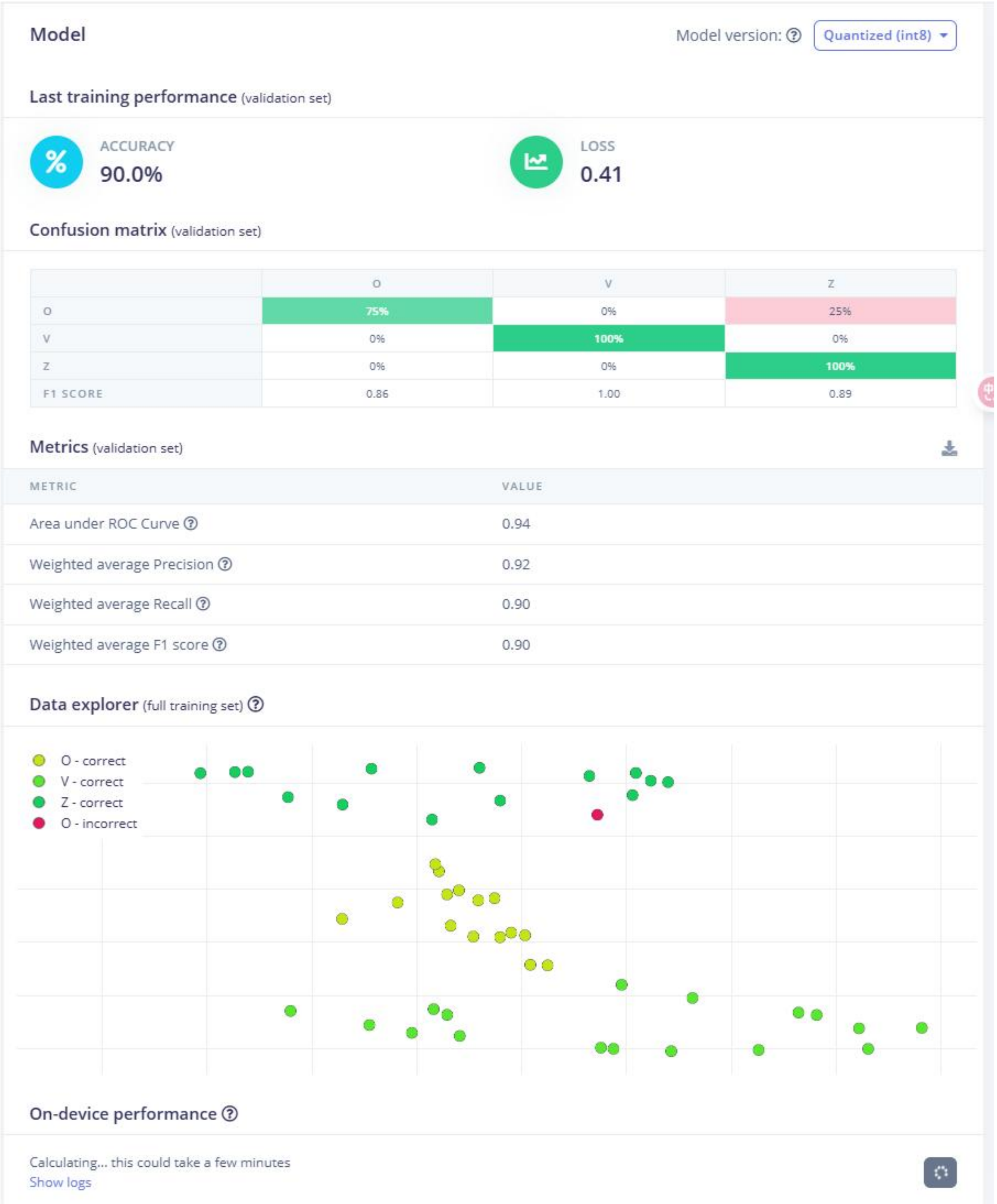


ML block **before** tuning: no good

This attempt didn't work well probably because the model was too small to capture differences between the three gestures.



ML block **after** tuning: good


Model
Model version: Quantized (int8)
Last training performance (validation set)
ACCURACY
90.0%
LOSS
0.41
Confusion matrix (validation set)

	O	V	Z
O	75%	0%	25%
V	0%	100%	0%
Z	0%	0%	100%
F1 SCORE	0.86	1.00	0.89

Metrics (validation set)

METRIC	VALUE
Area under ROC Curve	0.94
Weighted average Precision	0.92
Weighted average Recall	0.90
Weighted average F1 score	0.90

Data explorer (full training set)

On-device performance
Calculating... this could take a few minutes
Show logs

To achieve better performance, I tuned the neural network with the following settings:

- Training cycles: 100
- Learning rate: 0.0005
- Architecture:
  - Input layer: 207 features
  - Hidden layers: 40 neurons → 20 neurons → 10 neurons

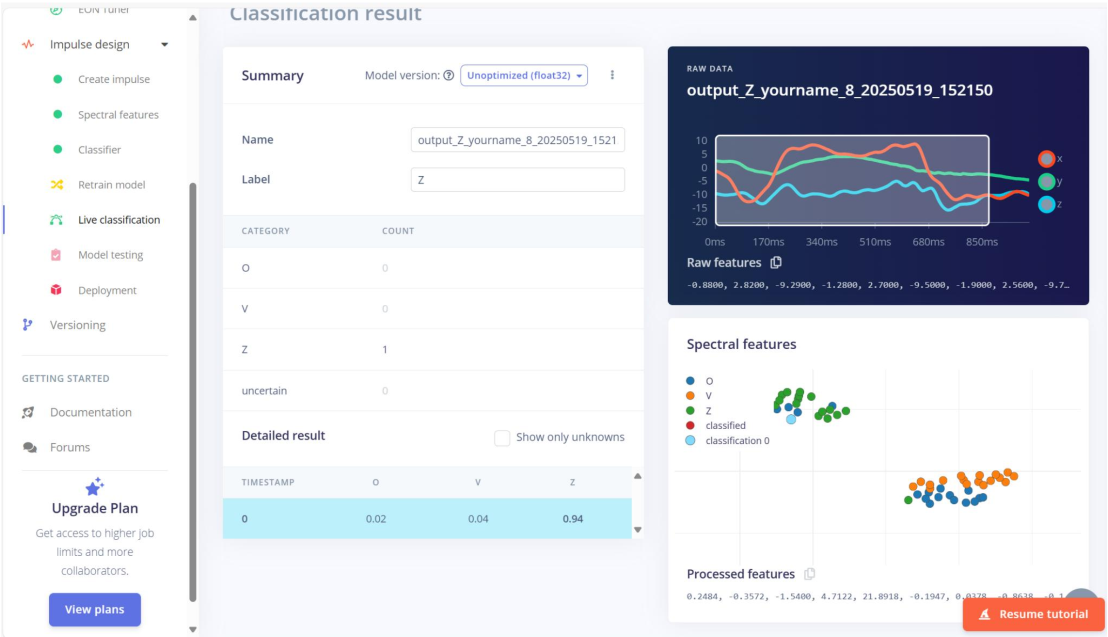
    - Output layer: 3 classes (O, V, Z)

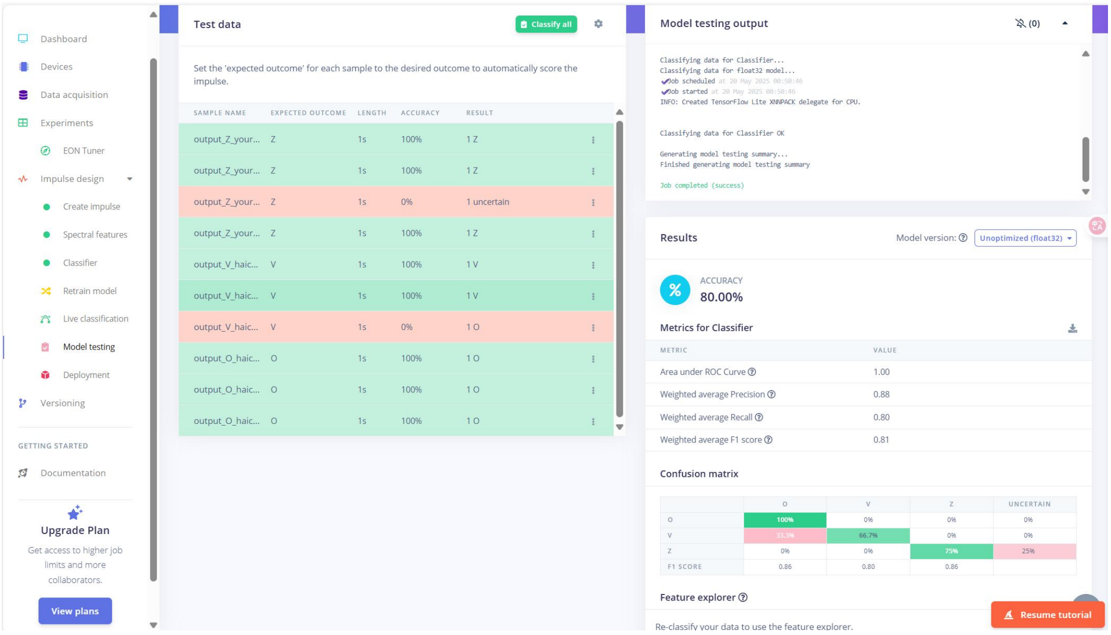
Training results:

- Accuracy: 90%
- Loss: 0.41
- F1 Scores: O: 0.86, V: 1.00, Z: 0.89
- Weighted average F1: 0.90
- Precision / Recall: 0.92 / 0.90
- AUC (ROC): 0.94

From the confusion matrix, we can see:

- Class V is perfectly recognized.
- Class Z has strong performance.
- Class O has 25% misclassified as Z, which is a minor weakness to improve.


Classification result
Impulse design
Create impulse
Spectral features
Classifier
Retrain model
Live classification
Model testing
Deployment
Versioning
GETTING STARTED
Documentation
Forums
Upgrade Plan
Get access to higher job limits and more collaborators.
View plans
Summary
Model version:
Unoptimized (float32)
Name
output_Z_yourname_8_20250519_1521
Label
Z
CATEGORY
COUNT
O 0
V 0
Z 1
uncertain 0
Detailed result
Show only unknowns
TIMESTAMP O V Z
0 0.02 0.04 0.94
RAW DATA
output_Z_yourname_8_20250519_152150
0ms 170ms 340ms 510ms 680ms 850ms
Raw features
-0.8800, 2.8200, -9.2900, -1.2800, 2.7000, -9.5000, -1.9000, 2.5600, -9.7…
Spectral features
O
V
Z
classified
classification 0
Processed features
0.2484, -0.3572, -1.5400, 4.7122, 21.8918, -0.1947,
Resume tutorial


Dashboard
Devices
Data acquisition
Experiments
EON Tuner
Impulse design
Create impulse
Spectral features
Classifier
Retrain model
Live classification
Model testing
Deployment
Versioning
GETTING STARTED
Documentation
Upgrade Plan
Get access to higher job limits and more collaborators.
View plans
Test data
Classify all
Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.
SAMPLE NAME EXPECTED OUTCOME LENGTH ACCURACY RESULT
output_Z_your... Z 1s 100% 1 Z
output_Z_your... Z 1s 100% 1 Z
output_Z_your... Z 1s 0% 1 uncertain
output_Z_your... Z 1s 100% 1 Z
output_V_haic... V 1s 100% 1 V
output_V_haic... V 1s 100% 1 V
output_V_haic... V 1s 0% 1 O
output_O_haic... O 1s 100% 1 O
output_O_haic... O 1s 100% 1 O
output_O_haic... O 1s 100% 1 O
Model testing output
Classifying data for Classifier...
Classifying data for float32 model...
Job scheduled
Job started
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Classifying data for Classifier OK
Generating model testing summary...
Finished generating model testing summary
Job completed (success)
Results
Model version:
Unoptimized (float32)
ACCURACY
80.00%
Metrics for Classifier
METRIC VALUE
Area under ROC Curve 1.00
Weighted average Precision 0.88
Weighted average Recall 0.80
Weighted average F1 score 0.81
Confusion matrix
O V Z UNCERTAIN
O 100% 0% 0% 0%
V 33.3% 66.7% 0% 0%
Z 0% 0% 75% 25%
F1 SCORE 0.86 0.80 0.86
Feature explorer
Re-classify your data to use the feature explorer.
Resume tutorial

Results

Model version: Unoptimized (float32)

ACCURACY
80.00%

Metrics for Classifier

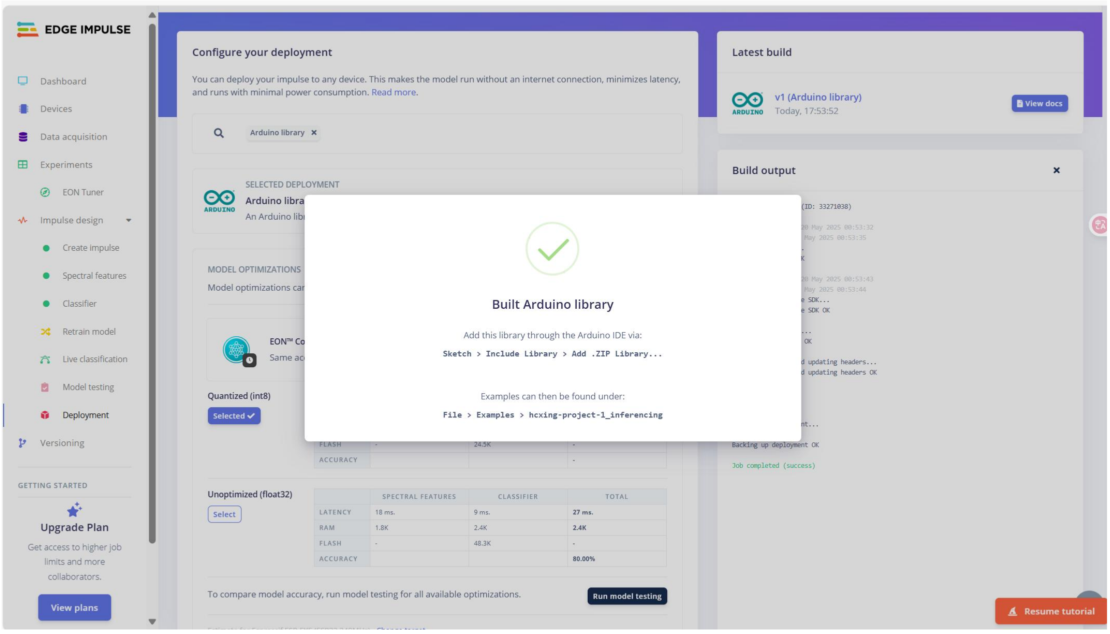
METRIC	VALUE
Area under ROC Curve	1.00
Weighted average Precision	0.88
Weighted average Recall	0.80
Weighted average F1 score	0.81

Confusion matrix

	O	V	Z	UNCERTAIN
O	100%	0%	0%	0%
V	33.3%	66.7%	0%	0%
Z	0%	0%	75%	25%
F1 SCORE	0.86	0.80	0.86	

Feature explorer

Re-classify your data to use the feature explorer.



"Live classification" and "Model testing":

- Accuracy: 80%
- Precision (Weighted average): 0.88
- Recall (Weighted average): 0.80
- F1 Score (Weighted average): 0.81
- Area under ROC Curve (AUC): 1.00
- Confusion Matrix:
  - Class O: 100% correct
  - Class V: 66.7% correct (33.3% misclassified as O)
  - Class Z: 75% correct (25% classified as uncertain)

Discussion: Give at least two potential strategies to further enhance your model performance.

1. Collect More and Higher-Quality Training Data

Increasing the number of well-labeled and consistently performed gesture samples—especially for underperforming classes like O—can help the model learn more robust patterns and reduce misclassification.

2. Optimize DSP and Neural Network Architecture

- DSP (Feature Extraction): Fine-tuning the window size and increasing frequency resolution may help extract more distinctive features for each gesture.
- Model Architecture: Adding more hidden layers or increasing the number of neurons can improve the model′s ability to capture complex motion patterns, though this must be balanced against on-device constraints.

4. Performance analysis and metrics

See details in answers for Q3

5. Answers to questions and your choices to all design options with justifications

See details in answers for Q3

6. Demo video link

https://drive.google.com/file/d/1j32liNdkkQn-Mwv-bW7rUMiV8yGIr32J/view?usp=sharing

7. Challenges faced and solutions

A major challenge I faced was getting the RGB LED to light up based on gesture recognition results. Even though the LED worked perfectly in basic test sketches—where red, green, and blue could all be lit using digitalWrite()—it consistently failed to light up in the integrated gesture recognition program. This issue blocked my progress for nearly two hours.

To debug the problem, I added detailed serial print statements to trace what value the prediction string actually held. I discovered that the predictions looked correct in the Serial Monitor (e.g., Prediction: O), but the LEDs still didn't respond. This led me to suspect that the string comparison using strcmp(prediction, "o") == 0 was failing silently.

To verify this, I printed the raw hexadecimal values of the characters in prediction, and noticed that they were uppercase (e.g., 'O' is 0x4F) while I was comparing against lowercase letters like "o". Additionally, there may have been invisible trailing characters or formatting from the classifier result that broke exact matches.

Solution:

I replaced all strcmp() comparisons with strstr() for partial matching (e.g., if (strstr(prediction, "O"))). This allowed the code to match substrings regardless of case or extra characters. After this change, the LEDs lit up correctly for each gesture prediction.

This bug was subtle because the Serial Monitor output looked correct, but the internal string content wasn’t exactly what I expected. Solving this helped me better understand how C-style string matching works on embedded systems, and how small mismatches can cause significant behavior issues.